Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Martina Gabor

Možnosti avtentikacije brez gesla z uporabo sistema Bitcoin

diplomsko delo

visokošolski strokovni študijski program prve stopnje računalništvo in informatika

Mentor: doc. dr. Mojca Ciglarič

Ljubljana 2015

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo: Možnosti avtentikacije brez gesla z uporabo sistema Bitcoin

Tematika naloge:

Naredite kratek pregled področja uporabe kriptografije za namen avtentikacije uporabnika. Predstavite klasične načine avtentikacije, ki se danes največ uporabljajo, ter za vsak način pojasnite področja uporabe ter prednosti in slabosti. Zlasti posvetite pozornost najbolj ranljivemu tipu avtentikacije – uporabniško ime in geslo. V nadaljevanju pojasnite, kdaj bi lahko bila avtentikacija brez gesla bolj varna in na katerih področjih bi bila njena uporaba bolj primerna. Preučite, kako bi lahko za avtentikacijo uporabljali mehanizme, ki jih nudi sistem Bitcoin. Predlog tudi implementirajte in demonstrirajte njegovo uporabo. Nazadnje kritično ovrednotite izvedbo in uporabnost svojega izdelka.

Izjava o avtorstvu diplomskega dela

Spodaj podpisana Martina Gabor sem avtor diplomskega dela z naslovom:

Možnosti avtentikacije brez gesla z uporabo sistema Bitcoin

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mojce Ciglarič,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 11. september 2015 Podpis avtorja:

Education is what remains after one has forgotten what one has learned in school.
Albert Einstein

Kazalo

Seznam uporabljenih kratic

IT Information technology, informacijska tehnologija

ID Identifier, identifikator

AAA Authentication, Authorization, Accouting, avtentikacija, avtorizacija, beleženje

PIN Personal Identification Number, osebna identifikacijska številka

SSO Single sign-on, osebna identifikacijska številka

p2p Peer-to-Peer, omrežje vsak z vsakim

BTC Bitcoin, elektronska valuta

PHP Hypertext Preprocessor, skriptni jezik za razvoj spletnih aplikacij

HTML Hypertext Transfer Protocol, protokol za izmenjavo hiperteksta

CSS Cascading Style Sheets, prekrivni slogi za urejanje spletne strani

SQL Structured Query Language, jezik za dostop do podatkov

JSON JavaScript Object Notation JavaScript, objektna notacija

RPC Remote procedure call, uporaba procedurj na oddaljenih računalnikih

API Application Programming Interface, programski vmesnik

URL Uniform Resource Locator, enolični krajevnik vira

Povzetek

V diplomski nalogi je na kratko predstavljeno področje avtentikacije in kriptografije za namen avtentikacije. Predstavljeni so klasični načini avtentikacije, ki so v današnjem času največkrat uporabljeni in velikokrat tudi najbolj ranljivi. Največkrat uporabljen način avtentikacije je z uporabniškim imenom in geslom. Ker za obisk raznih spletnih strani potrebujemo vedno novo geslo pride do prenasičenosti s številnimi gesli, ki pa so zaradi številčnosti slaba in posledično ranljiva. Da bi odpravili te težave, je bila razvita spletna aplikacija, ki omogoča avtentikacijo brez gesla z uporabo Bitcoin sistema. Uporabnik se avtenticira na podlagi bitcoin naslova. Z implementirano rešitvijo se uporabnika razbremeni gesel, omogočena pa je tudi določena stopnja anonimnosti pri prijavi v sistem.

Ključne besede: avtentikacija, bitcoin, geslo.

Abstract

The present diploma paper discusses the field of authentication and cryptology as a tool for authentication. The mostly used classic ways of authentication are presented, which are often also the most vulnerable ones. The mostly used way of authentication is the one with a username and a password. As people are required to create new passwords for different webpages, they are soon sated with the big amount of them, while the passwords get bad and vulnerable, because of the great number people have to remember. To solve this problem, a web application has been developed which enables authentication without a password, by using the Bitcoin system. The users authenticate themselves with the Bitcoin address. Thus the user does not have to remember a great number of passwords anymore, while a certain amount of anonymity at login is also possible.

Keywords: authentication, bitcoin, password.

Poglavje 1

Uvod

Hiter razvoj informacijskih tehnologij (IT) je poskrbel, da so se le-te zelo približale splošnim uporabnikom, pravzaprav je razvoj povzročil odvisnost družbe od IT. Z razvojem pametnih telefonov, ki so trenuten trend v svetu, s praktičnimi zasloni na dotik in povezavo s spletom, se je odvisnost še dodatno povečala. S tem smo dobili možnost, da glede na to, da imamo telefon s sabo povsod, lahko od kjerkoli in kadarkoli dostopamo do želenih podatkov, ki jih imamo v telefonu ali pa jih preverimo na spletu. Pregledamo lahko spletno pošto, preberemo novice, obiščemo družabna omrežja idr. Dostopati do velike večine aplikacij ne moremo, dokler ≫jim≪ ne povemo, kdo smo in dokler nam aplikacije dejansko ne verjamejo, da smo res oseba, za katero se izdajamo. Torej, za dostop do aplikacije je potrebno iti skozi proces avtentikacije, ki po navadi poteka z uporabniškim imenom in geslom. Če se držimo pravil varnega gesla, moramo za vsako izmed aplikacij imeti različno geslo (spletna pošta, Facebook, Twiter, dostop do sistema ...).

Prenasičenost z gesli je v današnjem svetu velika težava. Ravno zaradi tega ustvarjamo gesla, ki so kratka, takšna, ki imajo za nas sentimentalno vrednost, so hitro zapomnljiva, enaka gesla uporabljamo na različnih aplikacijah. Skratka, v veliki meri uporabljamo gesla, ki so preprosta. Preprosta predvsem za napadalce, ki se našega gesla želijo polastiti in ga uporabiti v nam škodljive namene.

Obstaja veliko programov, ki so namenjeni hranjenju gesel. Ob uporabi takšnega programa si je potrebno zapomniti samo geslo, s katerim dostopamo do programa, vprašanje pa je, ali mu zaupamo dovolj, da mu povemo vsa naša gesla. Kaj pa, če bi se, namesto da se ukvarjamo s tem vprašanjem, geslom preprosto izognili.

Predvsem zaradi prenasičenosti je želja, kako se izogniti kopici gesel, ki za splošnega uporabnika predstavljajo problem, prerasla v idejo. Kako ustvariti avtentikacijo, ki ne bo uporabljala gesla, uporabnik, bi pa za vstop v določen sistem potreboval samo uporabniško ime (ID). To bi uporabnika zagotovo razbremenilo in mu omogočilo boljšo uporabniško izkušnjo s katerimkoli sistemom. V izogib odkrivanju tople vode pa bi se to implementiralo s pomočjo infrastrukture in virov, ki že omogočajo in imajo implementirane določene funkcionalnosti.

V ta namen smo se odločili pregledati trenutne načine avtentikacije in alternativne ter implementirati obrazec za alternativno avtentikacijo brez gesla, katerega ID bo temeljil na odprtokodnem sistemu Bitcoin.

Kljub temu, da se geslo za avtentikacijo ne uporablja, uporabnik sistem za avtentikacijo vseeno potrebuje identifikator (avtentikacija po principu "nekaj, kar imaš"). V ta namen se uporabi bitcoin naslov, ki se brezplačno pridobi v elektronski denarnici.

Poglavje 2

Pregled področja avtentikacije in osnovni pojmi

V poglavju bomo podrobneje pregledali področje avtentikacije. Pregledali bomo obstoječe načine, njihove prednosti in slabosti. Podrobneje bomo pregledali tudi področje kriptografije, ki nastopa pri avtentikaciji.

2.1 Avtentikacija

Osnova za računalniško varnost je nadzorovan dostop do sistema, kar pomeni, da je nekdo pooblaščen, da dostopa do sistema. Da je dostop res lahko nadzorovan, moramo biti prepričani, da je nekdo, ki dostopa do sistema, res ta, za katerega se izdaja. Tak proces imenujemo avtentikacija. Z avtentikacijo preverjamo identiteto uporabnika, ki nam jo največkrat posreduje v obliki uporabniškega imena in gesla. S pravilnostjo teh nas uporabnik prepriča, da je na drugi strani res tisti, za katerega trdi, da je. [16]

Po avtentikaciji sledi tako imenovana avtorizacija, ki uporabniku podeli določene pravice za dostop do storitev. S tem določimo, kaj uporabnik na sistemu lahko počne in česa ne sme. Po avtorizaciji pa nastopi proces beleženja. To pomeni, da spremljamo statistiko seje določenega uporabnika.

Procesi avtentikacije, avtorizacije in beleženja skupaj nastopajo pod kra-

tico AAA, ki predstavlja varnostni arhitekturni model za nadzor nad uporabniki. Predmet diplomske naloge je avtentikacija, zato se bomo v nadaljevanju podrobneje posvetili le temu delu.

Avtentikacija temelji na nečem, kar poznamo, smo (fizična prisotnost) ali pa nečem, kar imamo. Avtentikacija mora biti zasebna in dovolj močna, da je lahko varna. [2]

2.1.1 Obstoječi načini

Poznamo tri vrste mehanizmov za preverjanje uporabnikove identitete:

- **Nekaj, kar poznamo:** geslo, PIN, skrivno vprašanje;
- **nekaj, kar smo:** prstni odtis, prepoznavanje glasa, razpoznavanje obraza;
- **nekaj, kar imamo:** ključi, kartica, vozniško dovoljenje, certifikat.

2.1.2 Nekaj, kar poznamo

Geslo je prva in najbolj razširjena ter poznana oblika avtentikacije. Uporaba gesel je enostavna in poceni. Uporabnik vnese uporabniško ime, ki je lahko javno poznano in se po navadi nanaša na uporabnikovo ime, spletno pošto ali psevdonim, zato zraven uporabniškega imena, zahtevamo še geslo. V primeru, da se vpisano geslo ujema z geslom, ki je shranjeno v sistemu za določenega uporabnika, se mu odobri dostop do sistema.

Težava pri uporabi gesel se pojavi, ker ljudje izbiramo preprosta gesla. Najslabša so tista, in posledično dobra za napadalca, ki so enaka uporabniškemu imenu ali so iz njega izpeljana, tista, ki imajo sentimentalno vrednost za uporabnika na primer ime hišnega ljubljenčka, otroka, drugega člana družine. Skrbno izbrano geslo je lahko močen avtentikator. [16]

Navodila za izbiro dobrega gesla:

- Uporaba vseh znakov in ne samo a-z ter malih in velikih začetnic.
- Uporaba dolgih nizov. Geslo naj je dolgo vsaj 8 znakov.
- Izogibajmo se imenom in besedam, ki so v slovarjih in imajo smisel. Uporabimo besede, ki so nesmiselne. Uporabimo stavek: "Danes je v Ljubljani lep in topel sončen dan!"in sestavimo geslo iz prvih črk: DjvLlitsd!.
- Pogosto spreminjajmo gesla. S tem preprečimo napadalcem, da bi imeli dovolj časa za ugotovitev gesla.
- Geslo mora ostati tajno, kar pomeni, da gesla nikomur ne zaupamo in ga ne pišemo na papir.

Vsako geslo je mogoče uganiti, močnost gesla je odvisna le od števila poizkusov, ki so potrebni za ugotovitev gesla. Močnejše kot je geslo, več poizkusov in posledično v povprečju (če ima napadalec srečo, lahko teoretično tudi dolgo geslo ugane v prvem poskusu) daljše časovno obdobje je potrebno za njegovo ugotovitev. [16]

Napadi na gesla

Ugibanje gesel

Zaradi šibkih gesel, ki jih izbiramo uporabniki, je le te mogoče ugotoviti z ustreznimi orodji oziroma programi, ki preizkušajo najpogostejša gesla, slovarske besede, osebna imena in podobno. Veliko uporabnikov ne menja privzetih gesel (user, guest, admin). Na spletu obstaja veliko seznamov takih gesel. Tako je možno enostavno, brez posebnih tehnik zgolj z ugibanjem ugotoviti gesla.

Napad z grobo silo (ang. brute force attack)

Velja za najpreprostejšo obliko napada na gesla. Pri tem napadu napadalec poskuša vse možne kombinacije znakov in jih vnaša v program. Pri kratkih geslih je ta tehnika zelo uporabna, saj lahko v sprejemljivem času ugotovi geslo.

Napad s slovarjem (ang. dictionary attack)

Pri napadu s slovarjem si napadalec za ugotovitev gesla pomaga s slovarjem. Napad temelji na seznamu besed iz slovarja (osebna imena, krajevna imena, pogosta gesla), ki jih vnaša kot geslo. Obstajajo programi, ki to delajo namesto napadalca in ki izdelajo za vsako besedo različne variante, ki vključujejo variacije malih in velikih črk ter podobno. Velja za najučinkovitejšo metodo, ki deluje neposredno na geslo.

Mavrične tabele (ang. rainbow table)

Pri tem napadu se izvede napad na shranjena gesla, kar pomeni, da napadalec ne poskuša uganiti gesla kakor pri prejšnjih primerih napadov. Gesla se v golem besedilu navadno ne shranjujejo, pač pa se hranijo njihove zgoščene vrednosti. Ko napadalec pridobi zgoščevalno vrednost, lahko ugotovi prvotno geslo, tako da poskusi vse možne kombinacije besed in nad njimi izvede zgoščevalno funkcijo ter vrednost, ki jo dobi, primerja z vrednostjo, ki je v datoteki. Ta napad je podoben napadu z grobo silo, le da pri tem primeru ne vnašamo vsakega gesla v program, ki preverja geslo, ampak imamo tabelo z že izračunanimi najpogostejšimi ali celo vsemi možnimi zgoščenimi vrednostmi. Iz tabele dobimo število, ki se zgosti v isto vrednost kot iskano geslo, kar zadošča za izvedbo napada. [9]

Načini za zmanjšanje števila gesel

Poznamo način avtentikacije uporabnika, ki omogoča prijavo do različnih aplikacij ali spletnih strani z enim uporabniškim imenom in geslom. To imenujemo enotna prijava (ang. Single Sign-On - SSO). Uporabnik se avtenticira le enkrat, nato pa mu vse storitve, dokler je prijava veljavna, dovolijo dostop, ne da bi zahtevale ponovno avtentikacijo. S tem se zmanjša število gesel, kar je osrednji problem našega diplomskega dela, saj uporabnik za vse uporablja eno geslo. Z uporabo tega načina pridobi boljšo uporabniško izkušnjo.

Današnja vodilna podjetja za spletne aplikacije kot so Facebook, Google, Yahoo, Twitter and PayPal uporabljajo storitve enotne prijave. V članku [25] so podrobneje preučeni varnostni problemi, ki se po navadi pri teh načinih pojavljajo. Kljub vsemu bi z implementacijo npr. Facebook enotne prijave v našo aplikacijo, delno rešili težavo, saj bi zmanjšali število gesel, vendar pa bi za avtentikacijo vseeno potrebovali geslo. Pa tudi uporabnik bi se moral registrirati in ustvariti račun na Facebooku. Zato vztrajamo naprej pri svoji rešitvi, za katero ni potrebno podati nobenih osebnih podatkov in ki ne potrebuje gesla. [25]

Skrito vprašanje

Pri tej vrsti avtentikacije nastopa tudi skrito vprašanje. Odgovor nanj bo vedela samo prava oseba. Vprašanja se navadno nanašajo na: mamino srednje ime, ime ulice iz otroštva, model prvega avtomobila, ime prvega hišnega ljubljenčka ali ime najljubšega učitelja iz otroštva. Uporabnik ob kreiranju identitete izbere želeno vprašanje in nanj poda odgovor. Vprašanje nam sistem zastavi na primer, ko pozabimo osnovno geslo in ga želimo poenostaviti. Celotna varnost temelji na predpostavki, da na skrivno vprašanje ne bo znal odgovoriti nihče razen nas (oziroma vsaj ne napadalec).

Težava pri skritih vprašanjih se pojavi, ker so lahko nekateri podatki uporabnika (npr. kraj bivanja, podatki staršev) javno objavljeni in dostopni.

Kar pomeni, da lahko na vprašanje pravilno odgovori vsak, ki odgovor pozna in ne samo uporabnik. [16]

2.1.3 Biometrična avtentikacija

Nekaj, kar smo ali biometrična avtentikacija, kar pomeni, da avtentikacija temelji na fizičnih karakteristikah uporabnika. Uporablja se že stoletja, vendar se komaj sedaj seli v računalništvo.

Prijatelje, znance, sorodnike prepoznamo po obrazu. Prav tako jih prepoznamo po tonu glasu preko telefona. Prepoznamo jih na fotografijah. Vse to temelji na biometrični avtentikaciji in jo uporabljamo dnevno.

Biometrična avtentikacija ima prednost pred ostalimi v tem, da ne more biti izgubljena, ukradena, pozabljena ali deljena z drugimi in je vedno dostopna, zaradi tega, ker temelji na človeških karakteristikah.

Slabosti

Zaradi precejšnje novosti na trgu je za nekatere kulture še vedno nesprejemljivo dajanje prstnega odtisa in drugih biometričnih lastnosti.

Prav tako so biometrične naprave trenutno še relativno drage in posledično nedostopne veliki večini uporabnikov. Ta težava se bo odpravila sčasoma z večjo priljubljenostjo in zanimanjem ter posledičnim znižanjem cen naprav.

Zaradi različnih dejavnikov v realnem življenju, lahko pride do ne prepoznavanja, zaradi spremembe človeških lastnosti. Stres lahko vpliva na naš glas, prst si lahko poškodujemo, lahko imamo suho kožo, položaj prsta na čitalniku lahko vpliva na branje prstnega odtisa, sprememba barve las, sprememba teže.

2.1.4 Nekaj, kar imamo

To pomeni avtentikacijo na podlagi fizičnega avtentitikatorja, ki ga imamo v lasti. Najbolj osnovni je ključ, s katerim se srečujemo dnevno in nam omogoča odklepanje vrat ter posledično vstop v želeni prostor. Seveda mora biti ključ ustrezen. Ne odklepa naš ključ vseh vrat. Ključ lahko izgubimo in v tem primeru lahko pride do zlorabe ključa in nekdo drug uporabi naš ključ.

Imamo tudi osebne izkaznice, razne kartice za popuste, ki jih uveljavljamo v trgovinah, potne liste ob vstopu in izstopu države ali na letalih, študentske izkaznice idr. S temi dokumenti nas ljudje prepoznajo in nam omogočajo določene dostope oziroma ugodnosti.

Poznane so tudi bančne kartice, ki vsebujejo magnetne čipe in razne kartice za dostope z brezžično tehnologijo. Kartico vstavimo v ustrezen čitalnik kartic in ta potem zazna kolikšna je vrednost na kartici.

Težava pri tej vrsti avtentikacije se pojavi, ko avtentikatorja nimamo pri sebi (pozabimo ga ali izgubimo), saj v tem primeru avtentikacija ni možna. [16]

2.2 Kriptografija

Kriptografija, beseda v grškem jeziku (kryptós - skrit in gráphein - pisati) pomeni skrito pisanje ali kriptologija (grško logos - vedenje) je veda, ki se ukvarja s tajnostjo, kriptiranjem in zakrivanjem sporočil ter razkrivanjem že kriptiranih podatkov (kriptoanaliza) s pomočjo matematičnih operacij. [14] Velja za najmočnejšo obrambo, ko gre za zaščito računalnikov. [2]

Uporaba kriptografije sega daleč v preteklost. Že v rimskem času jo je uporabljal Julij Cezar pri državniških opravkih. Svojim vojskovodjam je pošiljal kriptirana sporočila. Vsako črko v sporočilu je zamenjal s črko, ki je v abecedi nekaj mest za njo. [16]

Tabela 2.1 prikazuje primer, kjer je n velikost abecede A (slovenska abeceda n=25), ključ, ki je število, nam pove, za koliko mest bomo premaknili

osnovno abecedo. V našem primeru je ključ 3, kar pomeni, da naredimo premik znakov za tri mesta v desno. Zakodiramo besedo avtentikacija in dobimo tajnopis: čazhrzlnčelmč.

Zamik abecede za k=3	potek kriptiranja
A: ABCČDEFGHIJKLMNOPRSŠTUVZŽ	M= AVTENTIKACIJA
f(A): ČDEFGHIJKLMNOPRSŠTUVZŽABC	Ef(M)= ČAZHRZLNČELMČ

Tabela 2.1: Cezarjeva šifra.

Kriptografijo uporabljamo vsakodnevno bodisi pri telefonskih klicih, plačevanju z bančno kartico ali pri dviganju denarja z bankomata. Uporabljamo jo tudi pri prijavi z geslom v računalnik. Uporabljamo jo tudi pri pošiljanju občutljivih podatkov preko nezavarovanega omrežja, da podatki ostanejo berljivi samo uporabniku, kateremu jih pošiljamo.

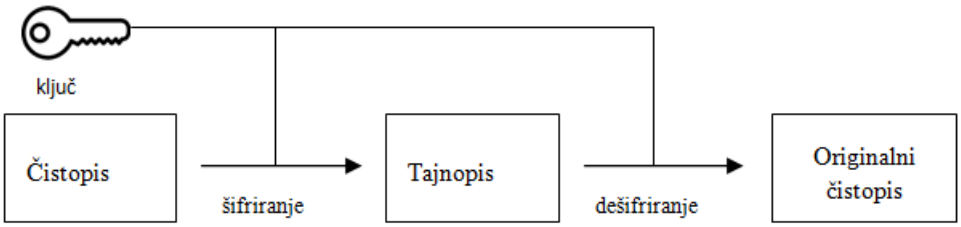
Pri kriptologiji uporabljamo termine enkripcija in dekripcija, kar pomeni kriptiranje in dekriptiranje podatkov. Prav tako uporabljamo pojme čistopis (cleartext, plaintext), ki predstavlja osnovno sporočilo ter šifropis ali tajnopis (ciphertext), kar pomeni zakriptirano sporočilo.

Kriptosistemi, ki skrbijo za kriptiranje sporočil, vsebujejo pravila, ki narekujejo, kako kriptirati oziroma dekriptirati sporočilo. Ta pravila imenujemo kriptirni algoritmi. Za parametre v algoritmu se uporabljajo vrednosti, ki jim pravimo ključi, kar pomeni, da se uporabnika za kriptiranje sporočila morata dogovoriti o algoritmu in ključu. Pri nekaterih vrstah kriptiranja se uporablja en ključ, ki je enak za kriptiranje sporočila in poznejše dekriptiranje. To imenujemo simetrična kriptografija. Poznana je tudi asimetrična kriptografija, pri kateri nastopata dva ključa. Eden za enkripcijo, drugi za dekripcijo. [13]

2.2.1 Simetrična kriptografija

Pri simetrični kriptografiji imata uporabnika, ki si izmenjujeta sporočila, kopijo enakega ključa. Tako en ključ uporabljamo za kriptiranje sporočila in prav tako za njegovo dekriptiranje. Ključ mora biti skriti in poznan samo uporabnikoma.

Pošiljatelj z algoritmom in ključem zakriptira osnovno sporočilo. Uporabnik, ki prejme sporočilo, uporabi obratni algoritem, kot ga je uporabil pošiljatelj z enakim ključem in tako dekriptira tajnopis, ki ga je prejel. Potek prikazuje slika 2.1.



Slika 2.1: Simetrična kriptografija.

Tako kriptiranje je hitro, težava pa je, kako varno izmenjati ključ. Uporabnika bi si morala ključ izmenjati preko varnega kanala, kar pomeni osebno, ali pa s kurirjem, ki se mu zaupa, kar pa je včasih težko ali pa celo nemogoče. Težava se pojavi tudi pri velikem številu ključev. Vsak par uporabnikov, ki si med seboj dopisujeta, mora imeti svoj ključ, kar pomeni, da ima en uporabnik lahko veliko število ključev za različne dopisovalce. Ravno zaradi tega so se razvile asimetrične metode, ki te slabosti odpravljajo.

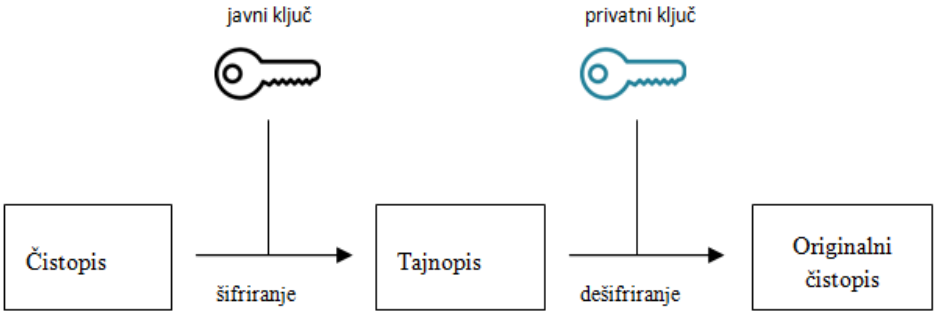
Najbolj poznana simetrična algoritma sta DES in AES.

2.2.2 Asimetrična kriptografija

Asimetrični kriptografiji pravimo tudi kriptografija javnega ključa. Vsak uporabnik ima dva ključa, javni ključ in pa zasebni, ki sta med seboj matematično povezana. Javni ključ se objavi kjerkoli, lahko na spletni strani ali

pa se pošlje po elektronski pošti. Zasebni ključ mora ostati skriti in poznan samo lastniku.

Pošiljatelj pridobi prejemnikov javni ključ. Sporočilo nato zakriptira z njegovim javnim ključem. Prejemnik prejeti tajnopis dekriptira s svojim zasebnim ključem, ki ga pozna le prejemnik sam. Potek prikazuje slika 2.2.



Slika 2.2: Asimetrična kriptografija.

Metode, ki se uporabljajo za računanje, so matematično bolj zahtevne kot pri simetrični kriptografiji, zato je ta metoda počasnejša.

2.2.3 Digitalni podpis

Digitalni podpis je protokol, ki ima enak učinek kot podpis na papirju. Je zaznamek, ki ga lahko naredi samo pošiljatelj, prejemnik pa ga zlahka poveže z njim. S podpisom potrjujemo podatke v sporočilu [16] in jih zavarujemo pred ponarejanjem. [10] Podpis jamči, da se podatki med pošiljanjem niso spremenili. [16]

Da ima digitalni podpis res enak učinek kot običajen, mora postopek zagotavljati:

- Avtentičnost,
- podpisa ni mogoče ponarediti, kopirali ali zanikati,

- podpisanega dokumenta ni mogoče spreminjati. [6]

Običajno se uporablja asimetrična kriptografija. Zaradi počasnosti pri velikih sporočilih, zaradi uporabe asimetrične kriptografije, se s pomočjo zgoščevalnih (hash) algoritmov naredi povzetek sporočila. Zgoščevalni algoritem vrne binarno vrednost fiksne dolžine (hash value), ki je povzetek vhodnega sporočila. Zgoščevalna funkcija je enosmerna, kar pomeni, da iz povzetka ni mogoče pridobiti vhodnega sporočila. Čeprav obstaja veliko različnih vhodnih sporočil, ki vrnejo isti povzetek, je težko (časovno zahtevno) za dano sporočilo najti še eno sporočilo, ki bo imelo enak povzetek. Koliko "težko"je to, je odvisno od posamezne zgoščevalne funkcije, pravzaprav od dolžine povzetka, ki ga funkcija vrača. [10]

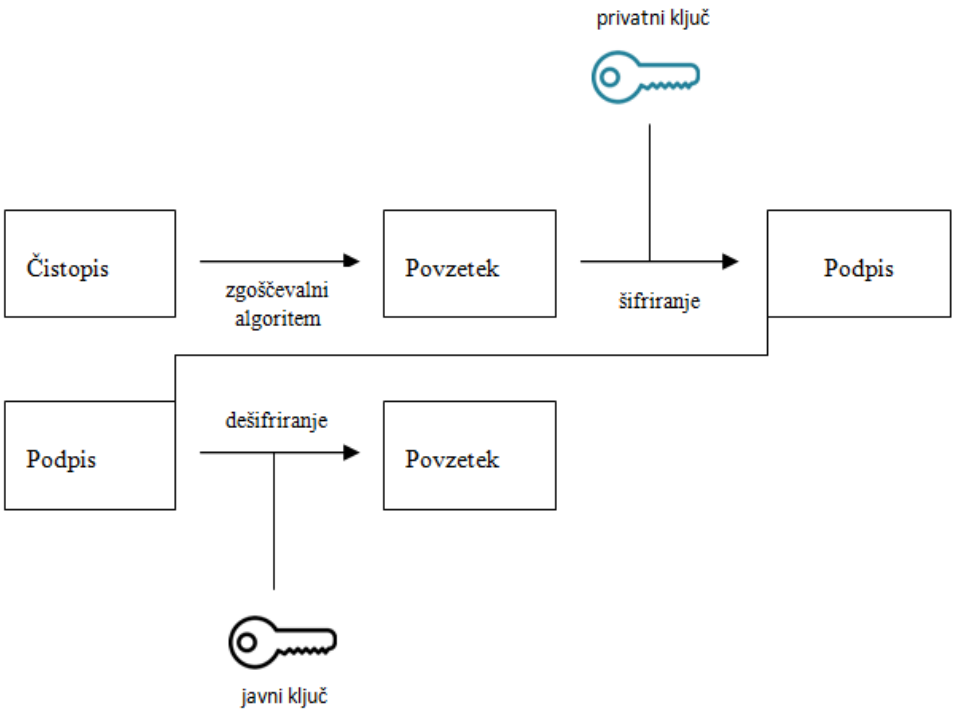
Pošiljatelj iz sporočila, ki ga želi prenesti do prejemnika z digitalnim podpisom, najprej izračuna povzetek s pomočjo zgoščevalnega algoritma. Povzetek nato zakriptira s svojim zasebnim ključem. Rezultat tega je podpis. Prejemnik podpis dekriptira s pošiljateljevim javnim ključem in dobi povzetek. Primerja povzetka in če se ujemata, se sporočilo med pošiljanjem ni spremenilo. Digitalni podpis lahko samo dodamo sporočilu ali pa ga vgradimo v sporočilo. Prikaz na sliki 2.3. [16]

2.2.4 Kriptirni algoritmi

Računalniški algoritmi poskrbijo, da je podatke, ki so kriptirani, težko prebrati. [16] Z njimi torej preslikamo čistopis v tajnopis. So matematične funkcije, namenjene kriptiranju in dekriptiranju, pri katerih je rezultat odvisen od vrednosti ključa. Pri različnih kriptirnih ključih se isti čistopis preslika v različne tajnopise. [10]

Obstaja veliko kriptirnih algoritmov, najbolj poznane smo našteli v tabeli 2.2 in jih tudi na kratko opisali.

- **DES** (ang. Data Encryption Standard): Je bil prvič objavljen leta 1977, ko so ga Združene države Amerike izbrale kot standard za po-



Slika 2.3: Digitalni podpis.

Simetrični	Asimetrični
AES	RSA
DES	ECC

Tabela 2.2: Kriptirni algoritmi.

datkovne komunikacije. Bil je najbolj uporabljen simetrični algoritem, dokler ni bil nad njim izveden napad z grobo silo. Zaradi tega več ne spada med varne simetrične algoritme.

- **AES** (ang. Advance Encryption Standrard): Je najbolj uporabljen simetrični algoritem, ki je nasledil algoritem DES, in je bistveno hitrejši ter varnejši. Temelji na operacijah nelinearne substitucije, transpozicije in mešanja stolpcev. [16]

- **RSA**: Je eden prvih in najbolj vsestransko uporabnih algoritmov javnega ključa. Generirali so ga leta 1978 Ron Rivest, Adi Shamir in Len Adleman, kratica RSA je sestavljena iz njihovih imen. Primeren je za kriptiranje in dekriptiranje ter podpisovanje. Uporablja se lahko kot podlaga za varno generiranje naključnih števil in za varnost v nekaterih elektronskih igricah. Njegova varnost temelji na zahtevnosti sestavljanja velikih celih števil. [2]

- **ECC** (ang. Elliptic curve cryptography): Asimetrični algoritem, ki temelji na eliptičnih krivuljah. V primerjavi z drugimi, ki ne temeljijo na eliptičnih krivuljah, zagotavlja enako stopnjo varnosti, s ključi manjše velikosti. Primeren je za enkripcijo, digitalni podpis in generiranje naključnih števil. [7]

Poglavje 3

Pregled sistema Bitcoin

Poglavje je namenjeno splošnemu pregledu sistema Bitcoin. Pojasnili bomo, kaj bitcoin je, kje se pridobi. Pojasnili bomo tudi pojme, ki se v povezavi z njim pojavljajo (denarnica, bitcoin naslov, transakcije).

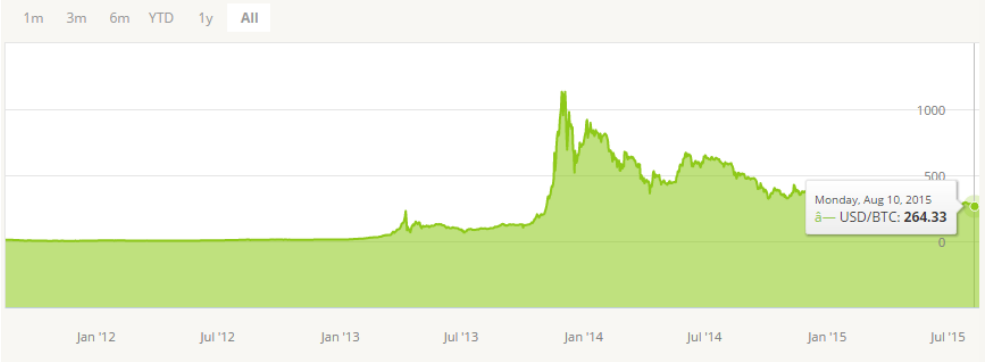
3.1 Bitcoin

Bitcoin (BTC) je decentralizirana, anonimna elektronska enota, tako imenovana kriptovaluta, implementirana na podlagi kriptografije in peer-to-peer (p2p) tehnologije. [15]

Bitcoine je možno pridobiti na spletnih borzah, kjer se zamenjajo za druge valute. Obstaja ogromno spletnih borz, ena bolj znanih je Bitstamp, ki sta jo ustanovila Slovenca, in omogoča enostaven nakup ter prodajo bitcoinov. Vseh bitcoinov še ni v obtoku, se pa pridobivajo s pomočjo postopka, ki se imenuje rudarjenje.

Vrednost bitcoina se neprestano spreminja, slika 3.1 prikazuje spreminjanje vrednosti BTCja skozi leta.

P2p plačilni sistem, ki je bil prvič objavljen leta 2008. V članku, avtorja, ki je želel ostati anonimen in se je podpisal s psevdonimom Satoshi Nakomoto, je bil opisal Bitcoin protokol, ki deluje na osnovi matematičnih pravil. Leta 2009 je postal odprtokodni. Ob povezavi z internetom in Bitcoin naslovom



Slika 3.1: Spreminjanje vrednosti BTC v dolarjih (Bitstamp).

lahko vsakdo po vsem svetu pošilja, ali sprejema Bitcoine. [22]

Tehnologija na kateri temelji Bitcoin je neodvisna in ne pripada nikomur, saj je celoten proces transakcij decentraliziran. Noben posameznik ali organizacija (osrednji organ ali banka) ne more manipulirati z algoritmom, zato ker je ta odprtokoden. Iz tega vidika je bitcoin mogoče obravnavati kot varen in neodvisen sistem. [8]

3.2 Bitcoin naslov

Bitcoin naslov je identifikator, ki vsebuje 34 naključnih alfanumeričnih latinskih znakov, razen številke 0 in črk O, I, i. Niz se začne z 1 ali 3. Številki predstavljata destinacijo za plačilo.

Vsakemu naslovu pripada javni in zasebni ključ ECDSA (protokol za generiranje digitalnih podpisov - Elliptic Curve Digital Signature Algorithm). Naslov je zgoščena vrednost javnega ključa. Uporabnik, ki želi poslati denar prejemniku, mora poznati le njegov javni ključ. Za razpolaganje z denarjem na določenem naslovu, pa je potrebno poznavanje zasebnega ključa. Če pride do izgube zasebnega ključa, lastnik Bitcoin naslova ne more več razpolagati s sredstvi na tem naslovu, kar pomeni, da so bitcoini na tem naslovu izgubljeni. [3] [4]

Uporabnik pridobi Bitcoin naslov z uporabo brezplačne programske opreme

na računalniku ali pametnem telefonu ali z uporabo spletnih storitev. V eni denarnici lahko ima uporabnik več naslovov, za boljšo varnost pri transakcijah, se za vsako priporoča nov naslov. Naslov se generira tako, da se vzame naključno število nad katerim se izvedejo matematične operacije, ki nam potem dajo javni in zasebni ključ. [22]

Primer naslova:

- 19VAVuupv9gbmySbKq9zUQzy6T297jsLoV.

3.3 Denarnica

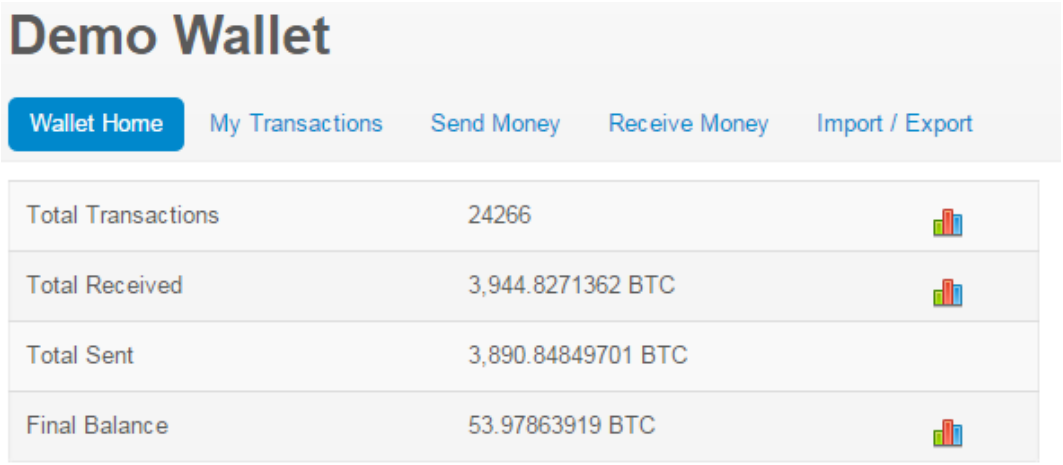
Denarnica (ang. wallet) je datoteka, ki vsebuje zasebni ključ, ki je namenjen generiranju javnega ključa in podpisovanju transakcij, preko katerih se pošiljajo bitcoini. Podpis se avtenticira z javnim ključem. Torej bitcoin uporablja kriptografijo javnega ključa, ki temelji na tem, da sta generirana javni in zasebni ključ.

Poznamo več vrst denarnic:

- **Spletna denarnica:** Denarnica se nahaja na spletu (Slika 3.2). Strežnik ponudnika storitve hrani naš ključ (preden se pošlje na strežnik, se kriptira).
- **Namizna denarnica:** Nameščena je na osebnem računalniku.
- **Mobilna denarnica:** Lastnik jo ima na svojem mobilnem telefonu. [3]

V denarnicah so informacije o trenutnem stanju v denarnici in podatki o vseh transakcijah, ki so vezani na naslove v denarnici. Vsebujejo tudi podatke o tem, koliko bitcoinov je bilo poslano oziroma prejeto v denarnico.

V denarnicah je omogočeno pošiljanje in prejemanje bitcoinov. Omogočajo tudi razne druge funkcionalnosti, kot na primer podpis poljubnega sporočila z zasebnim ključem denarnice, kot smo to mi uporabili pri implementaciji svoje



Slika 3.2: Primer demo denarnice na blockchain.info.

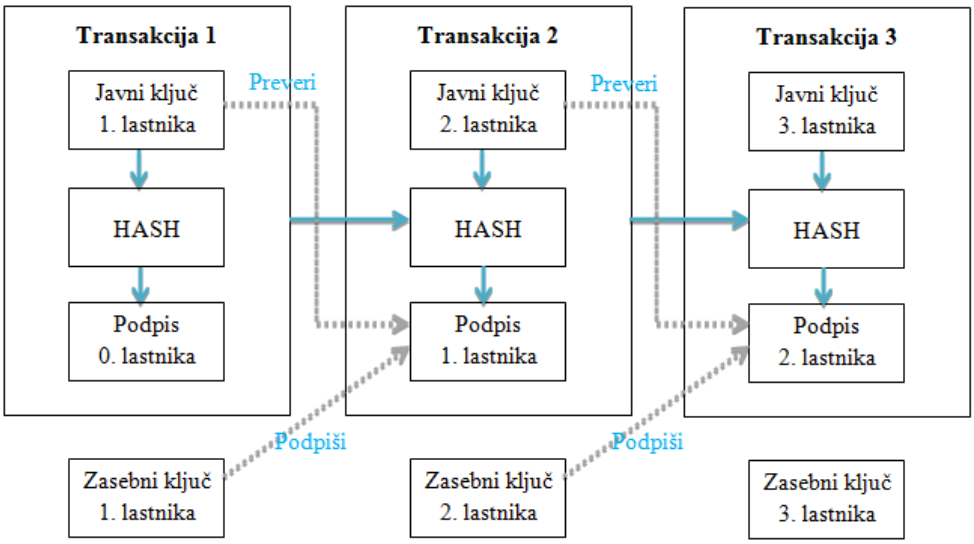
rešitve. Kakor že omenjeno omogočajo tudi kreiranje poljubnega števila naslovov za razpolaganje z bitcoini. [3]

3.4 Transakcije

Omogočene so transakcije med naslovi v omrežju, ki so zaradi varnosti digitalno podpisane. So kriptografsko podpisani zapisi, ki pošljejo bitcoine iz enega naslova na drugega.

Odjemalec uporabnika, ki pošilja BTC, sestavi transakcijo s podatki o prejšnji transakciji, s katerimi so bili BTC-ji prejeti na račun, količino BTC in naslovom, na katerega se pošiljajo. Transakcija se nato digitalno podpiše z uporabnikovim zasebnim ključem in se razpošlje v omrežje. Celoten potek dobro prikazuje slika 3.3.

Vse transakcije v omrežju se hranijo v javni glavni knjigi oziroma v verigi blokov (ang. blockchain). S tem je možno preverjati, da ne pride do podvajanja transakcij. Če pride do tega, obvelja tista, ki je bila v omrežju objavljena prej. Transakcije so dokončne, kar pomeni, da jih ni mogoče več preklicati, kakor smo vajeni pri standardnih načinih plačevanja. [3]



Slika 3.3: Potek transakcij bitcoinov. Vsak lastnik bitcoina pošlje tega tako, da podpiše zgoščeno vrednost prejšnje transakcije in javni ključ naslednjega prejemnika ter to doda na konec bloka.

3.4.1 Potrjevanje transakcij in rudarjenje

Da ne pride do dvojnega zapravljanja bitcoinov s strani uporabnikov, sistem uporablja decentralizirano potrjevanje transakcij. Transakcije se zapišejo v blok, ki se mu dodajo podatki o zadnjem predhodno potrjenem bloku, dodatne kontrolne informacije ter nekaj naključnih števil (nonce). Ta naključna števila mora rudar (ang. miner, razloženo v nadaljevanju) uganiti, da nato izračunana zgoščena vrednost po SHA256 (zgoščevalna funkcija), ustreza predpisanim pogojem. Če ustreza pogojem, je blok veljaven in se javno objavi, transakcije v bloku pa so potrjene. [23] Tak postopek se imenuje rudarjenje (njegovi udeleženci pa rudarji) in je namenjen obdelavi nakazil, varovanju omrežja in usklajenosti vseh udeležencev bitcoin sistema. [5]

Predstavili bomo primer rudarjenja na besedni zvezi "Hello,world!". Naš cilj je, da uganemo naključna števila, ki so potrebna, da se izračunana zgoščena vrednost po SHA256, začne z "0000". To se, kakor že omenjeno,

doseže tako, da se dodajajo naključna števila na konec niza. V spodnjih alinejah smo na nekaj primerih dodali naključne številke in prikazali, kakšna zgoščena vrednost se iz niza in naključnih števil pridobi. Rešitev smo našli pri številu 4250.

- "Hello, world!0":
  1312af178c253f84028d480a6adc1e25e81caa44c749ec81976192e2ec934c64.
- "Hello, world!1":
  e9afc424b79e4f6ab42d99c81156d3a17228d6e1eef4139be78e948a9332a7d8.
- "Hello, world!2":
  ae37343a357a8297591625e7134cbea22f5928be8ca2a32aa475cf05fd4266b7.
- "Hello, world!4248":
  6e110d98b388e77e9c6f042ac6b497cec46660deef75a55ebc7cfdf65cc0b965..
- "Hello, world!4249":
  c004190b822f1669cac8dc37e761cb73652e7832fb814565702245cf26ebb9e6.
- "Hello, world!4250":
  0000c3af42fc31103f1fdc0151fa747ff87349a4714df7cc52ea464e12dcd4e9 [21]
  (pravilna rešitev, saj ustreza začetnim pogojem, da se niz začne z "0000").

Rudarjenje je tako, zaradi ugibanja naključnih števil, računsko zahteven proces. Opravlja se na specializirani strojni opremi, kjer je veliko poceni elektrike, zaradi zahtevnosti procesa, ki povzroča veliko porabo časa in električne energije. [23]

Rudarjenje je torej operacija, s katero se oblikujejo bloki za vpis v glavno knjigo. Rudarji, ki najdejo omenjene zgoščene vrednosti, so nagrajeni s fiksnim številom bitcoinov ter s provizijami vseh transakcij, ki so v izračunanem bloku. [3]

Ko se bo doseglo maksimalno število bitcoinov (končno število vnaprej določeno in omejeno na 21 milijonov, da se zagotovita predvidljivost in stabilnost valute), torej se bo ustavilo nastajanje novih, bo rudarjenje postalo neprofitno, saj rudarji ne bodo več nagrajeni za nove bitcoine. Takrat bodo rudarji lahko nagrajeni samo s provizijami od transakcij, ki jih bodo potrdili, za kar pa prejmejo bistveno manjšo nagrado kot pri ustvarjanju novih bitcoinov. [24]

Poglavje 4

Načrt in implementacija

V tem poglavju bomo v uvodu predstavili smiselnost svoje aplikacije v nadaljevanju pa orodja, tehnologije in programske jezike, ki smo jih uporabili pri implementaciji, ter razvoj aplikacije. Prikazali bomo tudi delovanje končne rešitve.

4.1 Smiselnost avtentikacije brez gesla

4.1.1 Prednosti

Da smo lastniki določenega bitcoin naslova, lahko dokažemo s tem, da neko transakcijo ali pa sporočilo podpišemo s svojim zasebnim ključem. Torej, samo oseba, ki ima zasebni ključ, lahko izvede podpis. Vsak, ki ima naš javni ključ (bitcoinov naslov) pa lahko verificira to transakcijo ali naš podpis določenega sporočila. Celoten bitcoinov sistem deluje tako, da tisti, ki ima pri sebi zasebni ključ, lahko podpiše sporočila za porabo bitcoinov. Enako je v našem primeru, ko podpisovanje uporabljamo za avtentikacijo. Kar zagotovo velja za prednost te uporabe, torej, samo tisti, ki ima zasebni ključ v lasti, ima pravico avtentikacije.

Kakor je bilo omenjeno že v uvodu, je glavni povod za razvijanje te rešitve prenasičenost z gesli. Takšna vrsta avtentikacije to težavo odpravi. Ustvariti si je potrebno le bitcoinov naslov, ki je samodejno generiran v elektronski

denarnici in hraniti zasebni ključ v denarnici na varnem.

Velika prednost rešitve je tudi v anonimnosti. Uporabnik ne potrebuje podati svojih osebnih podatkov, da bi opravil avtentikacijo, kar pomeni, da se od uporabnika zahteva zgolj njegov bitcoin naslov, ki predstavlja le niz 34 alfanumeričnih latinskih znakov. Čeprav velja, da so vse bitcoin transakcije javno in trajno objavljene v omrežju, kar pomeni, da lahko vsakdo spremlja vsa nakazila in stanja na kateremkoli bitcoin naslovu, identiteta lastnika bitcoin naslova ostaja skrita.

Povzetek prednosti:

- Samo lastnik zasebnega ključa, lahko opravi avtentikacijo.
- Avtentikacija samo z avtomatsko generiranim bitcoin naslovom.
- Bitcoin omogoča zakrivanje identitete uporabnika. Anonimnost.

4.1.2 Uporaba

Uporaba tovrstne avtentikacije pride v poštev tako pri spletnih aplikacijah kot pri aplikacijah v realnem življenju. Smiselna uporaba je povsod, kjer osebni podatki uporabnika niso pomembni ali pa jih želimo skriti.

Razni forumi ne potrebujejo osebnih podatkov, le identifikator, zato so primerni za uporabo. Tudi spletne trgovine, predvsem tiste, ki imajo implementirano plačilo z BTC, saj v tem primeru spletna trgovina ne potrebuje nobenih drugih podatkov, le naslov za dostavo je seveda potrebno vnesti. Tudi wiki strani ne potrebujejo osebnih podatkov in bi tako bile primerne.

V realnem življenju bi tako avtentikacijo lahko uporabili pri nadzoru dostopa do vrat. Bodisi v osebne stanovanjske prostore ali pa v hotelske. V teh imajo odpiranje sob navadno urejeno s ključi ali pa kartico, z implementacijo naše avtentikacije bi to lahko umaknili, saj bi vsak uprabnik imel omogočen dostop kar preko denarnice na pametnem telefonu, ne nazadnje imamo telefon skoraj vedno pri sebi. Enako bi lahko zadevo implementirali tudi pri

dostopu do raznih omaric, ki jih najdemo bodisi v nakupovalnih središčih, v fitnes centrih ali pa v kopališčih.

Primeri uporabe:

- Forumi,
- wikiji,
- spletne trgovine,
- nadzor dostopa do vrat (hoteli),
- ključavnice omaric.

4.1.3 Varnostni pomisleki

Če se bitcoin uporablja pravilno, omogoča visoko stopnjo varnosti. Uporabniki smo sami odgovorni za varnost svoje elektronske denarnice. Enako kot pri fizični denarnici, v kateri prenašamo gotovino. Ustvarjati moramo varnostne kopije denarnice. Varnostna kopija nas ščiti pred računalniškimi okvarami in raznimi človeškimi napakami. V primeru, da je kriptirana, nam omogoča tudi obnovitev denarnice v primeru kraje. [5]

- Odporni smo na napade, ki nas pestijo pri avtentikaciji z geslom.
- Uporabniki moramo biti pozorni na URL, ki se nam generira kot sporočilo, ki ga je potrebno podpisati (aplikacija generira sporočilo, ki je sestavljeno iz URLja spletne strani in iz naključne številke), da je res URL naše aplikacije, zaradi tega, da se izognemo napada "man in the middle", kjer bi nam lahko napadalec podtaknil svoje sporočilo.
- Uporabniki smo sami odgovorni za varovanje zasebnega ključa in ustvarjanje varnostnih kopij. Namreč glavna slabost avtentikacije je v izgubi zasebnega ključa, saj v tem primeru avtentikacija več ni možna.

4.2 Tehnologije in programski jeziki

Spletno rešitev smo razvijali na osebnem računalniku, na katerem smo postavili lokalni strežnik, v kombinaciji s PHP, MySql in PHPMyAdmin. To nam je služilo za sprotno testiranje HTML datotek z vključeno kodo PHP in datoteko za oblikovanje CSS, za upravljanje z bazo MySQL, preverjanje delovanja strežnika in za sprotni pregled spletne rešitve. Najbolj razširjen spletni strežnik za PHP je Apache. V ta namen smo uporabili paket XAMPP, ki vsebuje vsa orodja, ki smo jih našteli zgoraj. Samo kodo smo pisali v tekstovnem urejevalniku Notepad++.

4.2.1 PHP

Je odprtokodni programski jezik, ki ga uporabljamo za razvoj dinamičnih spletnih strani. Primarno teče na spletnem strežniku, kjer za vhod jemlje PHP izvorno kodo, spletno stran pa generira kot izhod. Podoben je običajnim strukturiranim programskim jezikom, najbolj jezikoma C in Perl. [19]

4.2.2 HTML in CSS

V osnovi HTML skrbi za strukturo oziroma vsebino spletne strani. Vsak HTML dokument se da prikazati na različen način z različnim oblikovanjem. Za oblikovanje spletne strani pa se uporablja CSS (Cascading Style Sheets). Vsebino dokumenta se lahko oblikuje tudi v samem HTML dokumentu, vendar je pri obsežnejših straneh to lahko časovno predolgo in posledično tudi predrago. Ravno v ta namen se je razvila slogovna predloga CSS, ki omogoča oblikovanje HTML elementov. [17]

4.2.3 MySQL

Je odprtokodna podatkovna baza, ki za delo s podatki uporablja jezik SQL. Sistem torej skrbi za upravljanje s podatkovnimi bazami. Deluje na principu odjemalec - strežnik. [18]

4.2.4 JSON-RPC PHP

Je zbirka razredov napisanih v PHP, ki imajo implementirane funkcionalnosti odjemalca in strežnika po protokolu RPC, ki za izmenjavo podatkov uporablja JSON. [12]

JSON je format zapisa za izmenjavo podatkov. Je enostaven za branje in pisanje ter razčlenjevanje in generiranje. JSON je idealen jezik za izmenjevanje podatkov, saj je neodvisen od programskih jezikov, vseeno pa uporablja določila, ki so sorodna C-družini programskih jezikov. [11]

Remote Procedure Call je tehnologija, ki omogoča klic metod, ki se ne nahajajo v naslovnem prostoru klicanega procesa. Metode se lahko nahajajo znotraj računalnika ali pa celo na internetu in se uporabljajo za medsebojno komunikacijo. Odjemalec potrebuje določeno storitev, pokliče metodo na strežniku, ki nudi to storitev in metoda vrne zahtevane podatke.

Za potrebe spletne rešitve je bil uporabljen dokument jsonRPCClient.php.

4.2.5 API

API je programski vmesnik sestavljen iz podprogramov, razredov, protokolov in orodij za izdelavo programskih aplikacij.

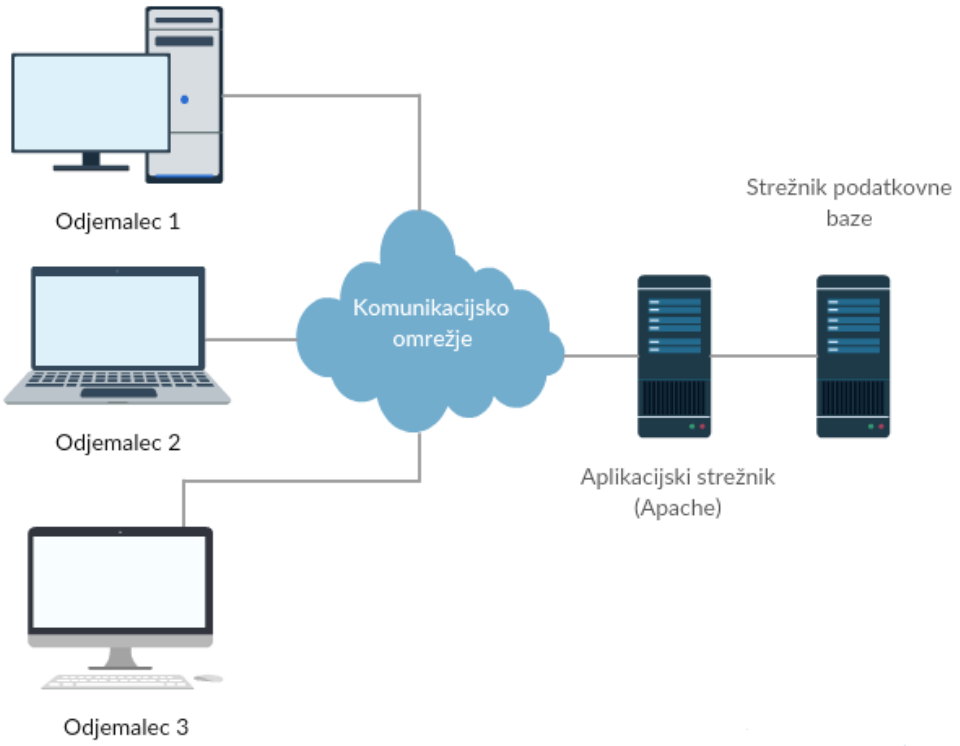
Za potrebe svoje avtentikacije smo uporabili API vmesnik spletne strani blockchain.info. Blockchain velja za najbolj priljubljeno Bitcoin denarnico z množico najbolj priljubljenih in uporabljenih programskih vmesnikov (API). Velja kot najmočnejša in najbolj zaupanja vredna znamka v Bitcoin industriji. [1]

API vmesnike smo uporabili zato, da nam metod, ki so že spisane, ni bilo treba ponovno pisati in smo jih klicali samo s pomočjo API vmesnikov.

4.3 Arhitektura

Aplikacija je razvita na principu arhitekture odjemalec - strežnik. Bolj natančno na principu 3-nivojske arhitekture (Slika 4.5):

- Odjemalec (1. nivo): izvaja uporabniški vmesnik in zahteva podatke.
- Aplikacijski strežnik (2. nivo): izvaja poslovna pravila in procesiranje.
- Strežnik podatkovne baze (3. nivo): shranjuje podatke, skrbi za integriteto in varnost.

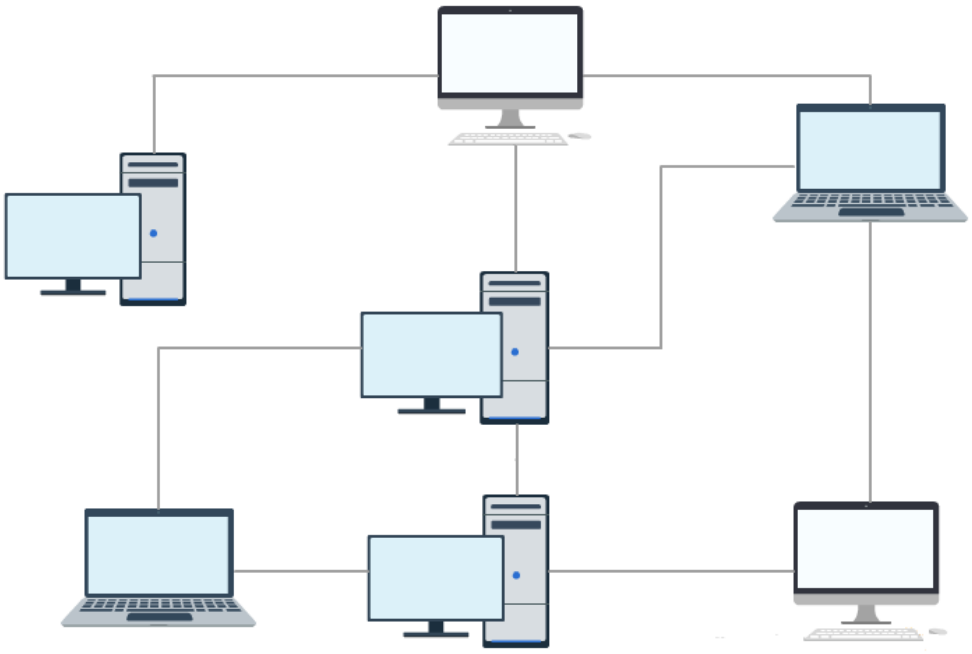


Slika 4.1: 3-nivojska arhitektura.

Za tako arhitekturo je značilno, da odjemalec izvaja malo ali celo nič procesiranja, le-to izvaja aplikacijski strežnik. Strežnik podatkovne baze pa preverja veljavnosti in vse dostope do podatkov.

Če primerjamo arhitekturo Bitcoin omrežja z našo aplikacijo, le-ta temelji na p2p arhitekturi (Slika 4.5). To pomeni, da lahko vsaka delovna postaja neposredno komunicira z drugo, brez posredovanja strežnika. Tako noben član sistema ni nadrejen oziroma podrejen, v smislu funkcij, ki jih opravlja

ali pa dostopa do podatkov, ki jih lahko vidi. Tak sistem je zato zelo robusten, ker lahko kateri koli član brez škode izpade in ni šibke točke. Medtem ko pri arhitekturi strežnik - odjemalec, če odpove strežnik, storitve, za katere je odgovoren, več niso zagotovljene in posledično aplikacija ne deluje.



Slika 4.2: P2P arhitektura.

Prednosti konfiguracije odjemalec - strežnik:

- Centralizirano upravljanje z viri,
- večja varnost in nadzor dostopa do datotek,
- zmanjšano upravljanje pri odjemalcih,
- možnost enostavnega povečanja sistema.

Slabosti konfiguracije odjemalec - strežnik:

- Višja cena (potrebno kupiti strežnike),
- v primeru ne delovanja strežnika, tudi omrežje ne deluje.

Prednosti konfiguracije p2p:

- Hitrejša dostava podatkov z enega računalnika na drugega,
- preprosta razširljivost,
- nudi večjo zasebnost.

Slabosti konfiguracije p2p:

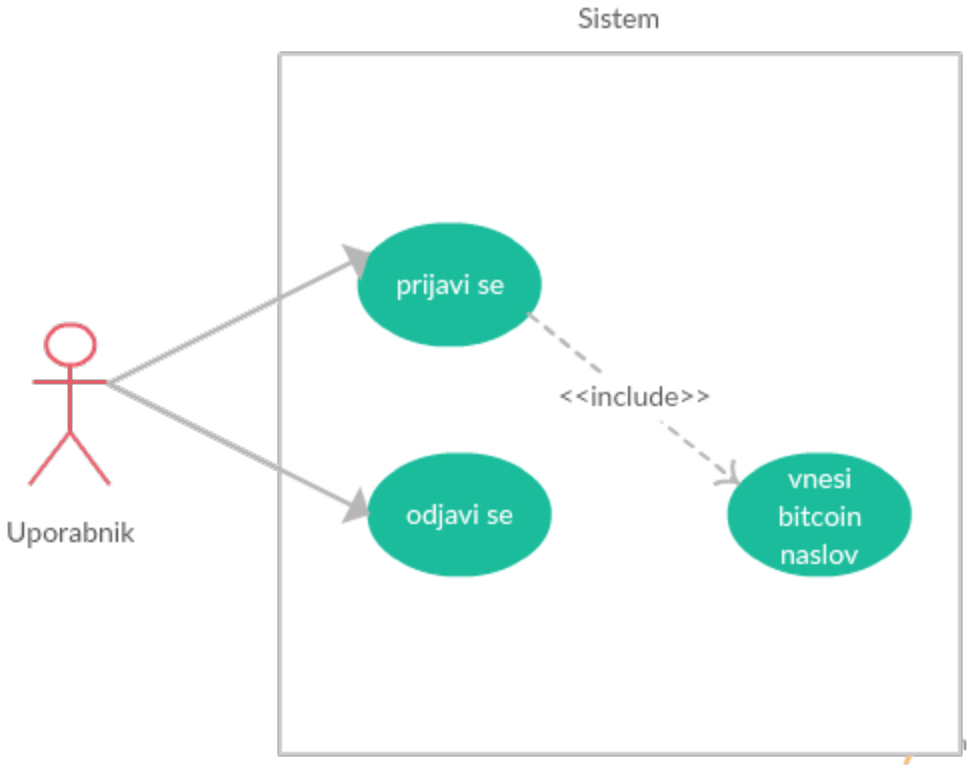
- Lahko se pojavijo težave z zaščito in nezanesljivostjo. [20]

4.4 Načrt

Preden smo se lotili implementacije naše rešitve, smo definirali funkcionalnosti, ki jih spletna aplikacija za avtentikacijo omogoča. Ta simulira prijavo uporabnika v aplikacijo brez gesla. Uporabniško ime predstavlja bitcoin naslov, alternativo geslu pa podpis samodejno generiranega sporočila, z zasebnim ključem, ki ga ima v lasti lastnik omenjenega bitcoin naslova. Na podlagi tega podpisa nas aplikacija avtenticira in opravi prijavo.

Po določitvi funkcionalnosti sistema smo izdelali diagram uporabe, ki ga prikazuje slika 4.3. Uporabnik ima omogočeno dejanje prijave, kjer mora vnesti bitcoin naslov in možnost odjave.

Izdelali smo tudi diagram zaporedja (ang. Sequence Diagram), ki je zelo uporaben za boljše načrtovanje določenega postopka znotraj aplikacije. Na sliki 4.4 smo prikazali, kako poteka prijava uporabnika na stran.

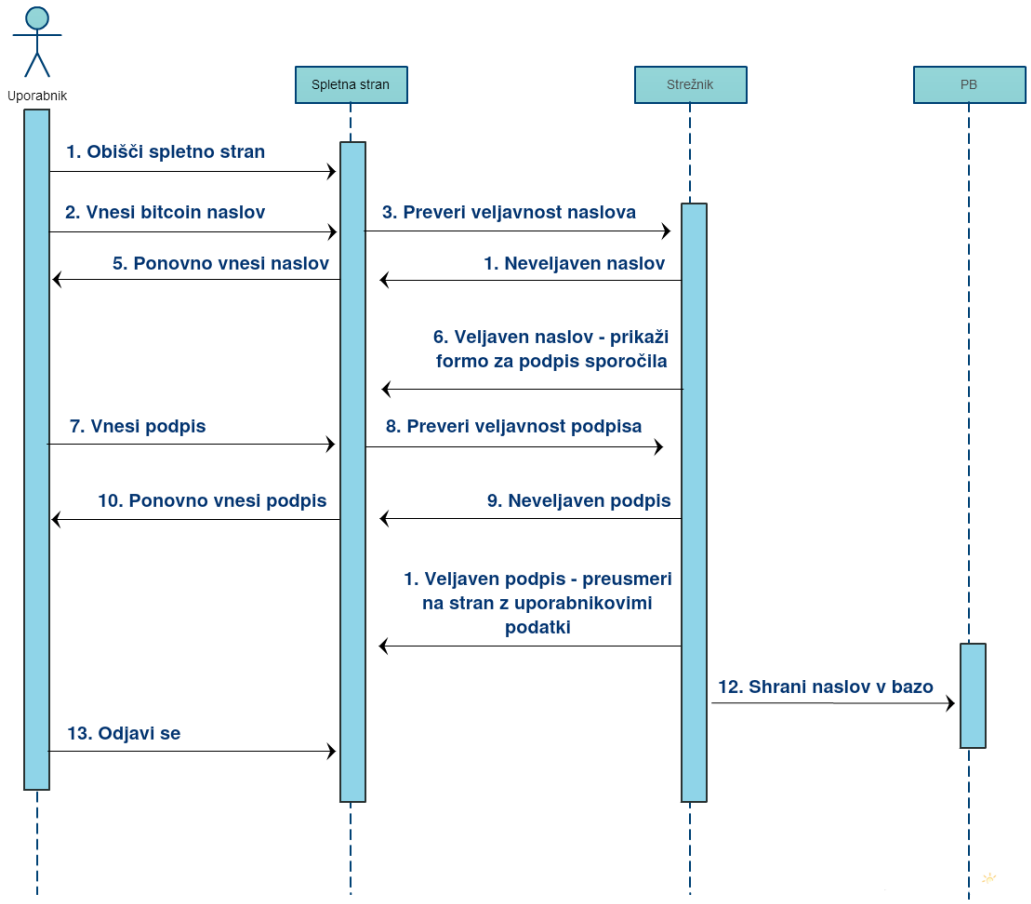


Slika 4.3: Primer uporabe.

4.5 Razvoj

Najprej smo pregledali sorodne rešitve, ki na trgu že obstajajo in katere so nam služile kot opora pri našem razvoju. Nadaljevali smo s pregledovanjem, kje in kako pridobiti podatke in metode, ki smo jih potrebovali za razvoj svoje rešitve. Ko smo vse to zbrali, smo se v nadaljevanju lotili programiranja spletne aplikacije.

Za zgoraj omenjene tehnologije in jezike smo se odločili na podlagi poznavanja le teh. Izbrali smo tiste, ki so nam bolj poznane in posledično lažje za delo. API vmesnik, ki smo ga uporabljali, je napisan za različne programske jezike, sami smo se odločili za uporabo PHP jezika zgolj iz razloga, ker je to široko uporabljan jezik za tovrstno spletno programiranje, poleg tega pa



Slika 4.4: Diagram zaporedja.

nam je sintaksa najbližja.

Za hranjenje podatkov o uporabnikih (beležimo: btc naslov in število prijav) smo izbrali MySQL podatkovno bazo.

Primer prikazuje vzpostavitev povezave do strežnika podatkovne baze in izbiro baze.

```
$server="127.0.0.1"; // ime streznika
$username="root"; // Mysql uporabnisko ime
$password=""; // Mysql geslo
$db_name="uporabnik"; // ime podatkovne baze
```

```
$tab_name="uporabnik"; // ime tabele

// povezava s streznikom in izbira baze
mysql_connect("$server", "$username", "$password")or die("
    Povezava ni mozna!");
mysql_select_db("$db_name")or die("Izbira PB ni mozna!");
```

Kakor že omenjeno smo uporabljali API programske vmesnike spletne strani blockchain.info. To nam je služilo za preverjanje veljavnosti bitcoin naslova in pa podpisa sporočila.

Spodnji del kode prikazuje vzpostavitev RPC klica za preverjanje veljavnosti podpisanega sporočila, ki kot parametre prejme uporabnikov btc naslov, njegov podpis in izvorno sporočilo, ki je bilo podpisano.

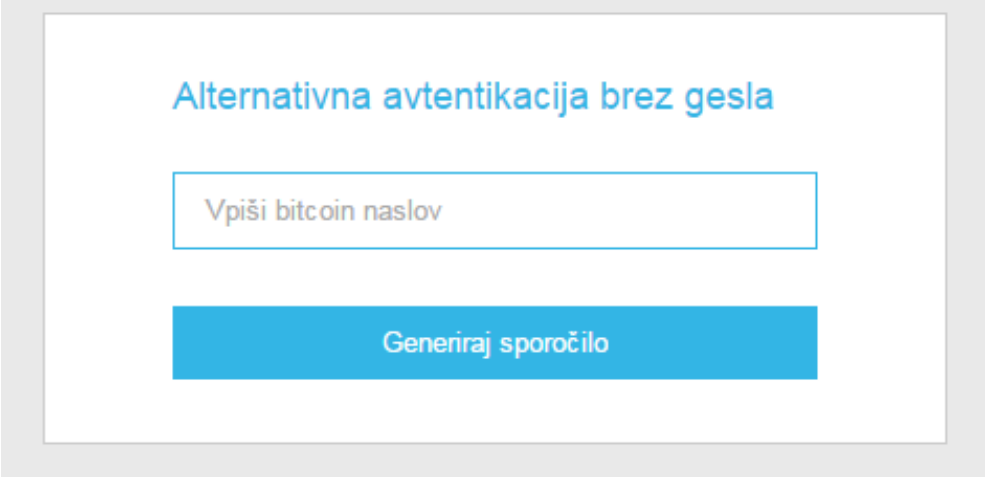
```
require_once 'jsonRPCClient.php';

$user = "e4948c7e-583f-4779-94a6-2b60cfa29c84";
$pass = "********";
$host = "blockchain.info";
$port = 443;

$rpc = new jsonRPCClient("https://{$user}:{$pass}@{$host}:{$port
    }");

//klicanje metode za preverbo veljavnosti naslova
$msgsigned = $rpc->verifymessage($myusername, $signature,
    $message);
```

4.6 Primer uporabe

4.6.1 Prvi korak

Prva stran spletne aplikacije vsebuje prijavno okno (Slika 4.5), v katerega smo podali veljaven bitcoin naslov. Ob kliku na gumb Generiraj sporočilo, se je s pomočjo PHP skripte, preverilo ali je podani naslov res veljaven. Nato smo

bili preusmerjeni na naslednjo stran. V primeru, da naslov ne bi bil veljaven, bi nas aplikacija na to opozorila in zahtevala, da naslov ponovno vpišemo. Seveda bi tudi nov naslov ponovno preverila ali je veljaven.



Slika 4.5: Začetna forma.

4.6.2 Drugi korak

Ko je bil bitcoin naslov sprejet, smo bili preusmerjeni na stran, ki je namenjena preverjanju veljavnosti podpisa primer na sliki 4.6. Aplikacija je generirala sporočilo, ki je sestavljeno iz URL aplikacije in na koncu dodane naključne devetmestne številke, ki ga je bilo potrebno podpisati. Podpiše se ga z zasebnim ključem lastnika bitcoin naslova.

4.6.3 Tretji korak

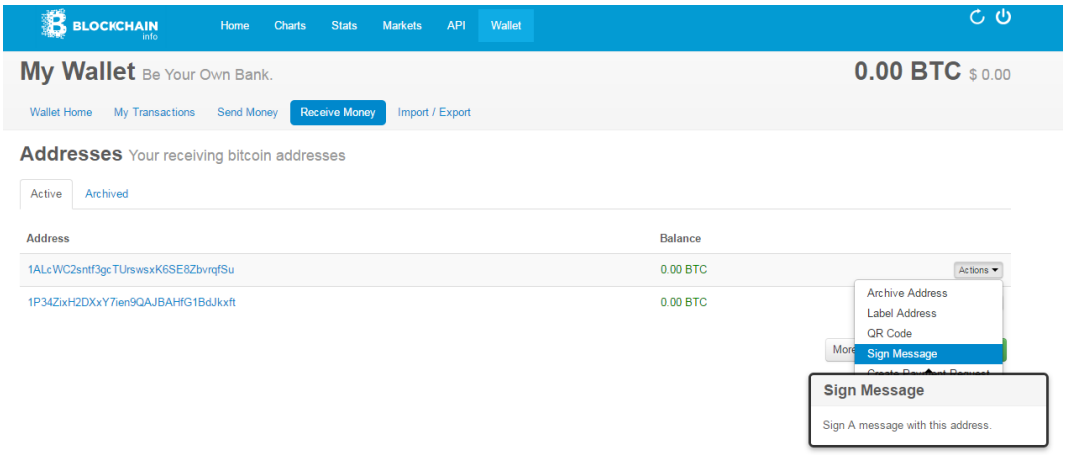
Zasebni ključ se nahaja v elektronski denarnici. Naša denarnico smo odprli pri Blockchain.info, zato smo nadaljevali na njihovi spletni aplikaciji. Aplikacija omogoča podpisovanje poljubnih sporočil, katerih namen je dokazovanje lastništva bitcoin naslova. Podpis sporočila smo izvedli s funkcijo, ki jo imajo implementirano v aplikacijo, Sign Message, kot prikazuje slika 4.7.

Slika 4.6: Preveri veljavnost podpisa.

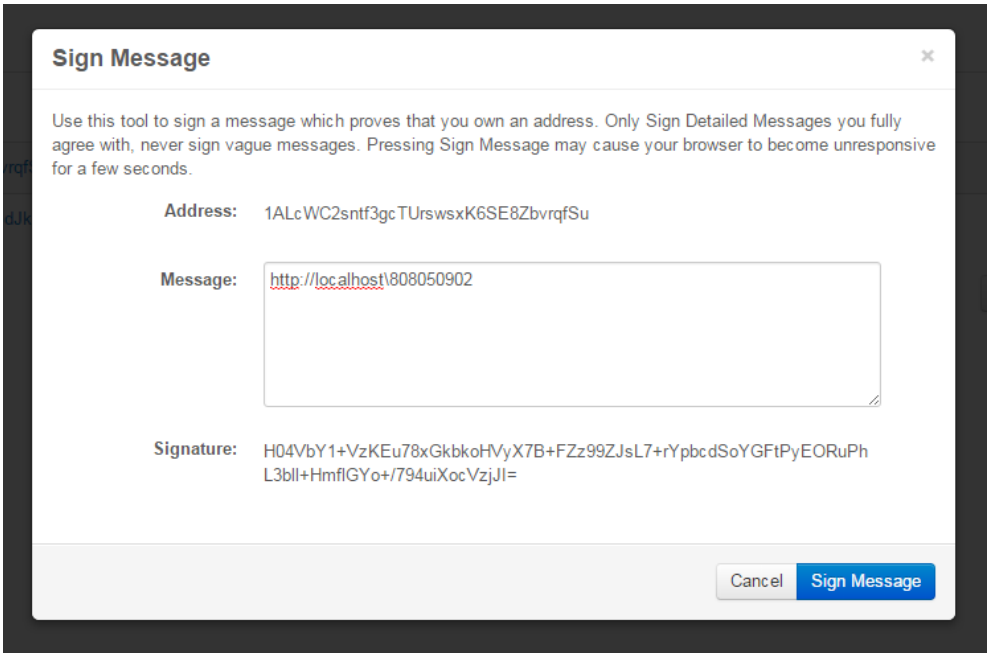
Funkcija je za parametre potrebovala naslov in sporočilo. Sporočilo smo kopirali s spletne aplikacije in kliknili gumb Sign Message, ki nam je nato vrnil podpis (Slika 4.8).

4.6.4 Četrti korak

Podpis smo skopirali in ga prilepili v svojo aplikacijo ter nadaljevali z gumbom Preveri veljavnost podpisa. S pomočjo funkcije Verifymessage, je bila preverjena veljavnost podpisa. Če je podpis veljaven se vzpostavi seja s trenutnim uporabnikom in se preusmeri na naslednjo stran, v kolikor pa ni veljaven se postopek validacije zahteva ponovno.



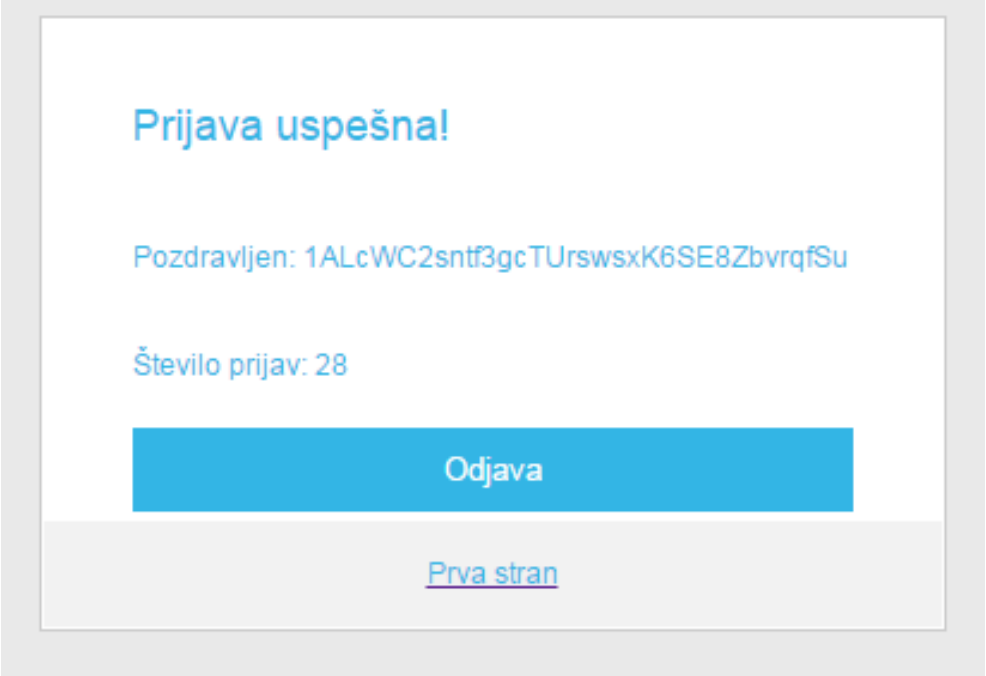
Slika 4.7: Funkcija za podpis sporočila.



Slika 4.8: Podpiši sporočilo.

4.6.5 Peti korak

Po uspešni verifikaciji, smo bili preusmerjeni na stran slika 4.9, ki vsebuje podatke o našem naslovu in pa število prijav v aplikacijo. Ima tudi gumb odjava, ki nas odjavi s seje in vrne na začetno stran.



Slika 4.9: Uspešna prijava.

Poglavje 5

Sklepne ugotovitve

Alternativa proti splošno znanim postopkom avtentikacije prinese določene prednosti. Predvsem odpravi preveliko število različnih gesel, kar je bil glavni problem diplomske naloge, in poenostavi postopek avtentikacije. S tem uporabnik pridobi boljšo uporabniško izkušnjo s sistemom, ki tovrstno avtentikacijo uporablja. Seveda pa alternativa prinese tudi nekatere slabosti, precej odgovornosti za varnost se prenese na uporabnika. Uporabnik mora sam poskrbeti, da njegovi zasebni ključi ostanejo na varnem, saj v nasprotnem primeru, izgubi možnost avtentikacije z naslovom, katerega zasebni ključ, je izgubil.

Ob raziskovanju dejstev smo naleteli na veliko razmišljanj, predvsem po raznih forumih, kjer ljudje želijo alternativne rešitve avtentikacije, ki bi odpravile težavo velikega števila gesel. To torej zagotovo je težava, na kateri se bo po našem mnenju vedno bolj delalo in bo mogoče sčasoma odpravljena. V bistvu se že dela na odpravljanju velikega števila gesel in sicer z enotnimi prijavami z identiteto, ki nam jo generira na primer Facebook ali Google.

Odkrili smo tudi nekaj primerov drugih implementacij, ki omogočajo avtentikacijo brez gesla, tako preko Bitcoin sistema, kot tudi na druge načine. Nobene rešitve pa nismo našli, ki bi bila uveljavljena v tej meri, da bi bila uporabljena na nam poznanih sistemih oziroma aplikacijah.

Zastavljeni cilj, implementacijo obrazca za alternativno avtentikacijo, smo

realizirali. Ob tem pa smo podrobneje spoznali avtentikacijski proces, kriptografijo in sistem Bitcoin.

5.1 Možne izboljšave

Izdelana rešitev je osnovna in predstavlja le prototip, katerega je možno še dodelati in na katerem se rešitev lahko razvija naprej.

Izboljšavo svoje implementacije vidimo v avtomatiziranosti procesa. To pomeni, da ročno kopiranje odstranimo in proces poenostavimo. Rešitev je za zdaj vidna v generiranju QR kode, preko katere bi se proces dalo avtomatizirati (trenutno je potrebno podpis sporočila ročno prekopirati na spletno stran). Uporabniku, bi se namesto sporočila genererirala kar QR koda, v katero bi sporočilo implementirali. Nekatere denarnice omogočajo skeniranje QR kode, uporabnik bi le-to poskeniral, ob tem bi se izvedla funkcija za podpis sporočila in povratni klic na spletno stran, ki bi avtenticiral uporabnika in mu omogočil dostop do želene strani. S tem bi proces avtentikacije poenostavili in pohitrili.

V diplomskem delu sem si zastavila in reševala problem prevelikega števila gesel, ki nam vsakodnevno povzroča preglavice. Pregledala sem načine avtentikacije, ki so že uveljavljeni in so v uporabi ter preučila možno rešitev mojega problema. Odločila sem se, da implementiram rešitev s pomočjo sistema bitcoin. Odločitev je temeljila na tem, da je sistem odprtokoden, decentraliziran in hkrati omogoča delno anonimnost uporabnika (vse transakcije so zabeležene v omrežju). Raziskala sem uporabljene tehnologije in ugotovila, da je rešitev izvedljiva. V nadaljnji fazi sem izdelalala načrt aplikacije. Na podlagi načrta, sem aplikacijo izdelala in testirala njeno delovanje. Na koncu sem ovrednotila nadaljnje možnosti za njen razcvet in razvoj.

Aplikacija ima možnost uveljavljanja predvsem pri uporabnikih, ki se želijo znebiti avtentikacije z geslom in prav tako pri uporabnikih, ki želijo biti anonimni pri uporabi spletnih aplikacij. Ravno anonimnost uporabnika

je dodana vrednost aplikacije. Pri uporabi določenih spletnih aplikacij ne želimo podajati svojih osebnih podatkov in želimo pri uporabi storitev ostati anonimni. Ta problem implementacija uspešno rešuje. Kakor je skozi diplomsko delo večkrat poudarjeno, je za avtenticiranje potreben samo bitcoin naslov, ki pa ne vsebuje naših osebnih podatkov, prav tako jih ne potrebujemo za njegovo pridobitev. Ravno v tem pa vidim tudi slabost implementacije na podlagi bitcoin sistema, saj se bitcoin omrežje šele dobro razvija in posledično še ni poznano velikemu krogu uporabnikov. Uporabnika bi bilo, v primeru, da ne pozna bitcoin sistema, pred uporabo tovrstne avtentikacije, potrebno seznaniti z delovanju bitcoin sistema. Če bi bitcoin postal valuta prihodnosti, bi se odpravila tudi ta slabost.

Literatura

[1] (2015) Application programming interface. [Online]. Dosegljivo: https://en.wikipedia.org/wiki/Application programming interface.

[2] C. Adams, S. Lloyd. Understanding PKI: Concepts, Standards, and Deployment Considerations, Addison Wesley, 2ns edition, 2002.

[3] (2015) Bitcoin. [Online]. Dosegljivo: https://sl.wikipedia.org/wiki/Bitcoin.

[4] (2015) Bitcoin address. [Online]. Dosegljivo: https://en.bitcoin.it/wiki/Address.

[5] (2015) Bitcoin. Kaj moram vedeti. [Online]. Dosegljivo: https://bitcoin.org/sl/kaj-moram-vedeti.

[6] (2015) Digitalni podpis. [Online]. Dosegljivo: http://www.si-ca.si/kripto/kr-podp.html.

[7] (2015) ECC. [Online]. Dosegljivo: https://en.wikipedia.org/wiki/Elliptic curve cryptography

[8] Eikmanns, Benedikt C. and Sandner, Philipp G., Bitcoin: The Next Revolution in International Payment Processing? An Empirical Analysis of Potential Use Cases (April 22, 2015).

[9] (2015) Gesla in napadi nanje. [Online]. Dosegljivo: http://www.monitor.si/clanek/gesla-in-napadi-nanje/122762/.

[10] Dušan Gleich, Žarko Čučej. Varnost informacij in omrežij : temelji na "Information and Network Security"od Miguel Soriano.

[11] (2015) Json. [Online]. Dosegljivo: http://json.org/.

[12] (2015) JSON RPC PHP. [Online]. Dosegljivo: http://jsonrpcphp.org/.

[13] Neal Koblitz , Alfred J. Menezes , Yi-Hong Wu , Robert J. Zuccherato, Algebraic aspects of cryptography, Springer-Verlag New York, Inc., New York, NY, 1998.

[14] (2015)Kriptografija. [Online]. Dosegljivo: https://sl.wikipedia.org/wiki/Kriptografija.

[15] J. A. Kroll, I. C. Davey, and E. W. Felten. The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries. In Proceedings of WEIS 2013, 2013.

[16] J. Margulies, S. L. Pfleeger, C. P. Pfleeger. Šecurity in Computing, Fifth Edition", 2015.

[17] J. Moffett, Chapter 4 - Why HTML and CSS, J. Moffett, Ed. Boston: Morgan Kaufmann, 2014.

[18] (2015) MySQL. [Online]. Dosegljivo: https://sl.wikipedia.org/wiki/MySQL.

[19] (2015) PHP. [Online]. Dosegljivo: https://sl.wikipedia.org/wiki/PHP.

[20] (2015) P2p. [Online]. Dosegljivo: http://im.scv.si/wiki/index.php/Peertoopeer.

[21] (2015) Primer rudarjenja. [Online]. Dosegljivo: https://en.bitcoin.it/wiki/Proof of work.

[22] Tan, Boon Seng and Low, Kin-Yew, Bitcoin: Its Economics and Financial Reporting (May 4, 2015). [Online]. Dosegljivo: http://ssrn.com/abstract=2602126.

[23] (2015) Težave pri rudarjenju bitcoinov. [Online]. Dosegljivo: http://www.monitor.si/novica/tezave-pri-rudarjenju-bitcoinov/167761/.

[24] François Velde. Bitcoin: a primer. Chicago Fed Letter, 2013. [Online] Dosegljivo: http://ideas.repec.org/a/fip/fedhle/y2013idecn317.html.

[25] Rui Wang; Shuo Chen; Xiaofeng Wang. Signing Me onto Your Accounts through Facebook and Google: A Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services. Security and Privacy (SP), 2012 IEEE Symposium on, strani: 365 - 379.